



DAT329

SQL Server 2005:

Index Maintenance Best Practices

Kimberly L. Tripp

President, SQLskills.com

Speaker – Kimberly L. Tripp

- Independent Consultant/Trainer/Speaker/Writer

- Founder, SYSolutions, Inc. www.SQLskills.com

email: Kimberly@SQLskills.com

Become a subscriber on SQLskills.com and learn about new resources which can improve your productivity and server performance!

- Microsoft Regional Director <http://www.microsoftregionaldirectors.com/Public/>

- SQL Server MVP <http://mvp.support.microsoft.com/>

- Author for some SQL Server 2005 Whitepapers on MSDN (links from www.SQLskills.com)

- Coauthor MSPress: SQL Server 2000 High Availability, Presenter/Technical Manager for SQL Server 2000 High Availability DVD

- Writer/Editor for SQL Magazine www.sqlmag.com

Overview

- Everything is about Availability
- How Fragmentation Occurs
- What Fragmentation Means
- How to See Fragmentation
- How to Minimize Fragmentation
- Rebuilding an Index
 - Why
 - When
 - How
- Automating Index Rebuilds
- Understanding Index Usage

Impacting Availability

- Hardware Failure...

- Clustering
- Log Shipping
- Database Mirroring

Chalk Talk (CHT056)

Achieving High-Availability with SQL 2005

Thursday, 7 July – 18:15-19:30

- Human Error...

- Database Snapshots
- Backups

Session (DAT315)

VLDB Availability and Recovery Strategies for SQL Server 2005

Thursday, 7 July – 12:00-13:15

- **Maintenance creates blocking...**

- **Online Index Rebuild**

How Fragmentation Occurs

- **Clustered Tables**

- **INSERTS:** Can be main problem if Primary Key is not monotonically increasing/decreasing.
- **UPDATES:** Depend on Row Definition and existence of variable width columns
- **DELETES:** Are a relatively minor problem compared to INSERTs/UPDATES as they leave gaps (depending on type of deletes). Gaps can be beneficial to OLTP as the freed space can be used for INSERTs/UPDATES. Range deletes where entire pages are “emptied” are freed and given back to the database for other uses.

- **Nonclustered Indexes**

- Most likely to have significant fragmentation as these are unlikely to ALL be created on ever increasing keys...

Index Fragmentation (Leaf Level)

- Data Modifications [can] lead to Fragmentation
- INSERT
 - Yes – Key value is not ever increasing/decreasing
 - NO – Key is ever increasing/decreasing
- UPDATE
 - Yes – Updates... to variable width columns – where the values are getting wider
 - NO – Columns are fixed width, columns have “place holder” values (i.e. DEFAULT constraints) to minimize row expansion on update OR no updates
- DELETE
 - Yes – Deletes are singleton deletes (swiss cheese problem)
 - NO – Deletes are RANGE deletes for archival purposes

How Fragmentation Occurs

Modifications Can Lead to Page Splits

- INSERTs to a page which is already full
(there is a special case when the INSERT is greater than the last row or less than the first row - on the page)
- UPDATEs to a variable width column on a page which is already full

**Data Page
Review and
Scenario**



How Fragmentation Occurs

Modifications Can Lead to Page Splits

- INSERTs to a page which is already full
(there is a special case when the INSERT is greater than the last row or less than the first row - on the page)
- UPDATEs to a variable width column on a page which is already full

**Data Page
Review and
Scenario**

Header 96 Bytes

**20
Rows/Page**

**8096
Bytes for**

**INSERT Customer
VALUES
('Atostle', ...)**

Albertson
Ambers
Anderson
...
Astronder

File1, Page
5982

Atkinson
Atonke
...
Baan
Bacen

File1, Page
5983

Bach
Badlen
Barber
...
Bendron

File1, Page
5984



How Fragmentation Splits Pages

If the page is full – on INSERT or UPDATE

1) A New Page is “linked” in

*Data Page
Review and
Scenario*
Header 96 Bytes

**20
Rows/Pag
e**
**8096
Bytes for**

**INSERT Customer
VALUES
(‘Atostle’, ...)**

Albertson
Ambers
Anderson
...
Astronder

File1, Page
5982

Atkinson
Atonke
...
Baan
Bacen

File1, Page
5983

Bach
Badlen
Barber
...
Bendron

File1, Page
5984

...



**New
Page
Linked
In**



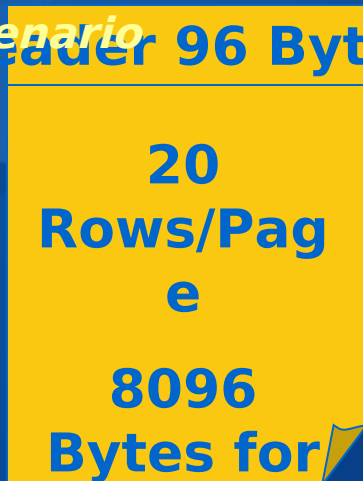
File1, Page
11231

How Fragmentation Splits Pages

If the page is full – on INSERT or UPDATE

- 1) A New Page is “linked” in
- 2) 50% of the page being split is moved to the newly linked in page

*Data Page
Review and
Scenario*
Header 96 Bytes



*INSERT Customer
VALUES
(‘Atostle’, ...)*



...



How Fragmentation Splits Pages

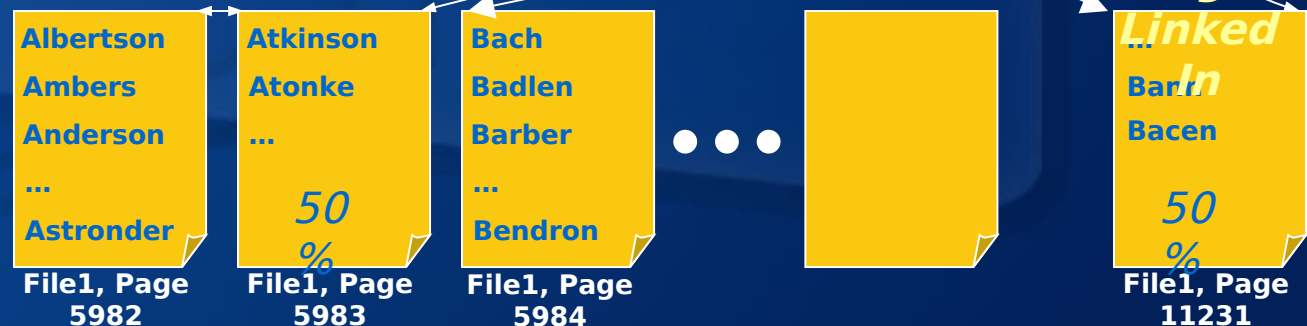
If the page is full – on INSERT or UPDATE

- 1) A New Page is “linked” in
- 2) 50% of the page being split is moved to the newly linked in page
- 3) Doubly-linked list/Pointers are updated to LOGICALLY follow the list

*Data Page
Review and
Scenario*
Header 96 Bytes



*INSERT Customer
VALUES
(‘Atostle’, ...)*



What Fragmentation Means

- Internal Fragmentation
 - Free Space on the Page
 - Desirable for OLTP
 - Free Space for New Records
 - Free Space for Updates
 - Undesirable OLAP
 - Table requires more pages
 - Large Analysis queries require more I/O



What Fragmentation Means

- External Fragmentation
 - Logical Left to Right Order of Leaf Level is NOT Physically Left to Right
 - Almost Always A Concern
 - Typically means splits have also resulted in internal fragmentation
 - Can cause disk thrashing for low cache and low selectivity queries



How to See Fragmentation

- Poor Query Performance over time
- More disk activity
- Poor Cache Utilization
- Verify Query I/O
 - SET STATISTICS IO ON
- Verify Scan Density
 - sys.dm_db_index_physical_stats
 - Manually
 - Programmatically
 - Automatically
- Periodically re-verify Query I/O

Detecting Fragmentation

- **SQL Server 2000: DBCC SHOWCONTIG**
 - Not tabular by default
 - Difficult to programmatically analyze
 - Pre-create temp table – have to know definition
 - Must use dynamic string execution to execute
- **SQL Server 2005:**
sys.dm_db_index_physical_stats(params)
 - Name length looks ugly...but MUCH better, dm objects make a large difference in overall server analysis
 - Table-valued function returns tabular set
 - EASY to programmatically analyze
 - Can use SELECT INTO to catch result set
 - No dynamic string execution needed!

SQL Server 2000

Resources

- Focus for this session – SQL Server 2005
- Concepts are the same, implementation is different
- Watch the MSDN Support Webcast: Indexing for Performance – Proper Index Maintenance (recorded 19 July 2004)

<http://msevents.microsoft.com/CUI/EventDetail.aspx?EventID=1032256511&Culture=en-US>

Analyzing Fragmentation

`sys.dm_db_index_physical_stats`

SQL Server 2005

- Query the new DMV: dynamic management view
- Multi-statement Table-valued function
- Can use SELECT INTO to create temp table for programmatic analysis (**great feature of ALL DMVs!**)
- Can run in multiple modes, target specific tables and/or indexes

```
SELECT *  
      INTO IndexFragmentation  
FROM sys.dm_db_index_physical_stats  
      (DatabaseID, TableID, IndexID,  
       PartitionNumber, Mode)
```

Analyzing Fragmentation

`sys.dm_db_index_physical_stats`

- `DatabaseID = [NULL | 'DatabaseID']`
 - `NULL`: returns information for ALL databases, if `NULL` is used no other options can be supplied. This returns ALL indexes for all objects in all databases. Easy but possibly slow.
 - `DatabaseID`: `smallint` type. Refers to the ID for a specific database. `DB_ID()` or `DB_ID('DatabaseID')` can be used. The latter allows you to run this from ANY database as long as you have access. However, 3-part naming must be used.

Analyzing Fragmentation

`sys.dm_db_index_physical_stats`

- `ObjectID = [DEFAULT | NULL | 'ObjectID']`
 - `DEFAULT/NULL`: return ALL base data: CL, Heap, LOB for the specified database.
 - `ObjectID`: int type. Refers to the ID for a specific object. `OBJECT_ID('TableName')` can be used. When using `OBJECT_ID`, you can use 1/2/3-part naming. Be sure to use 3-part when executing outside of database.
- `IndexID = [DEFAULT | NULL | 'IndexID']`
 - `DEFAULT/NULL`: All indexes
 - `IndexID`: tinyint type. Refers to the ID of a specific index.

Analyzing Fragmentation

`sys.dm_db_index_physical_stats`

- PartitionNumber = [DEFAULT | NULL | #]
 - DEFAULT/NULL/0: return ALL partitions
 - #: returns only the details about specific partition. When a PartitionNumber is specified then an IndexID must also be specified.
- Mode = [DEFAULT | NULL | 'SpecificMode']
 - DEFAULT/NULL/LIMITED: return FAST scan and use only an IS (Intent Shared) Table-level lock. This lock blocks ONLY eXclusive TABLE-level locks and schema changes. Excellent relatively unobtrusive way to get fragmentation details.

Analyzing Fragmentation

`sys.dm_db_index_physical_stats`

Mode (Default = Limited)

- **LIMITED: IS Lock.** Same as SQL 2000 WITH FAST, only page counts and EXTERNAL fragmentation displayed. Does not detail INTERNAL fragmentation and page density.
- ☑ **SAMPLED: IS Lock.** For tables less than 10,000 pages (~80MB), all details are produced. For tables of more than 80MB, two samples are done (1% and 2%) at every nth page. The samples are compared and if close, 2% sampling output returned. If not close, then up to 10% will be sampled.

Analyzing Fragmentation

`sys.dm_db_index_physical_stats`

- **DETAILED: S Lock.** Entire table analyzed for both internal and external fragmentation. Returns one row for each level of the index from the leaf level (level 0) all the way up to the root level. This can help you determine fragmentation in the non-leaf levels but at the expense of holding a shared table level lock.

NOTE: If fragmentation in the leaf level is minimized through proper index maintenance and fillfactor – then fragmentation in the non-leaf levels should be low.

Analyzing Fragmentation

`sys.dm_db_index_physical_stats`

- Using DMV determine if the indexes should be rebuilt or just defragmented
- General recommendations are for least impact and best performance results
- **Rebuild when:**
 - You have a regular and dedicated maintenance window
 - *Or* when AvgFragmentation > 30% and you can rebuild ONLINE
 - *Or* at the best possible time that has the least impact on production
 - if OFFLINE rebuild is required
- **Defrag when:**
 - AvgFragmentation <= 30%
 - You cannot rebuild online...

Analyzing Fragmentation

- External Fragmentation

- avg_fragmentation_in_percent (*should be low*)
 - Uses a percentage to show extent switches
- fragment_count (*should be low*)
 - How many “chunks” exist within db
- Avg_fragment_size (*should be high*)
 - Number of pages (avg) per fragment

- Internal Fragmentation

- Average Bytes Free Per Page
 - Amount of free space (if OLTP +, if OLAP -)
- Average Page Density (fullness)
 - Shown as a percentage – how FULL are the pages

Index Maintenance

- Rebuilding

- Rebuilds entire index structure
- Updates statistics
- Might require locks (if so, downtime)
- If required, lock types for rebuild:
 - Rebuild CL requires Exclusive Table Lock
 - Rebuild NC requires Shared Table Lock

- Defragging

- Not as exhaustive as rebuilding, only cleans up fragmentation in the leaf level
- Does not move the object for better extent scan density
- Always ONLINE (no long running locks)

Rebuilding an Index

- Generally better results w/rebuild v. defrag
- Completely removes all levels of fragmentation – from both the leaf level as well as the b-tree (i.e. completely rebalances the tree)
- Completely updates statistics – with the equivalent of a “full scan” (i.e. accurate statistics)
- **If offline, requires Locks ⇒ downtime**
 - Rebuild CL requires Exclusive Table Lock
 - Rebuild NC requires Shared Table Lock
- **How?**
 - ALTER INDEX... REBUILD (offline OR online – *preferred*)
 - DBCC DBREINDEX (offline only – *backward compatibility only*)
 - CREATE w/DROP_EXISTING = ON
(can change the CL index definition)

Defragging an Index

- Does not completely rebuild ALL levels of index – focuses on LEAF level
- Does not update statistics
- Does not require locks for length of transaction – defrag executes as mini-trans
- May take longer if table is extremely fragmented
- May take less time if table is not fragmented
- Does not run as a single large transaction – transaction log backups can execute and the transaction log space can be freed WHILE running
- How?
 - ALTER INDEX... REORG (*preferred*)
 - DBCC INDEXDEFRAG (*backward compatibility only*)

How to Minimize Fragmentation

- Add Internal Space on Pages to Allow Fluctuation in Row Size (due to updates against variable width columns)
- How?
 - When **rebuilding** set appropriate FILLFACTOR Setting
- Other Data Dependant Options?
 - Use Only Fixed Width Columns
 - No Updates
 - No Deletes

FILLFACTOR

- FILLFACTOR impacts ONLY the LEAF level of an index (with PAD_INDEX affects B-tree)
 - 0 – Default value w/special meaning. Leaf level is filled to 100%
 - Excellent for OLAP
 - Not ideal for OLTP
 - 100 – Leaf level is filled to 100% (same as default of 0)
 - Excellent for OLAP
 - Not ideal for OLTP
 - 1-99
 - IDEAL for OLTP – you know your data!
 - But how do you OPTIMALLY set this?

FILLFACTOR

Estimating appropriate FILLFACTOR

- What is going to cause splits in this structure?
 - UPDATES to variable width datatypes?
 - How wide?
 - How volatile?

The wider and/or more volatile ▯ **LOWER FillFactor**
- INSERTs?
 - How often?
 - Is the data ever-increasing?

The more volatile ▯ **LOWER FillFactor**
- How often can you rebuild?
 - The more frequent ▯ **HIGHER FillFactor**

FILLFACTOR

Testing your FILLFACTOR Estimate

- SELECT FROM sys.dm_db_index_physical_stats between rebuilds to see how far and how fast the scan density drops
 - The faster the table becomes fragmented (i.e. the lower the scan density goes)
 - ▮ LOWER FillFactor
 - or DECREASE the time between defrag/rebuilds
- Programmatically analyze the output from DMV to determine if a rebuild or defrag should occur

Restructuring an Index for more optimal Performance...

- Why?
- When?
- How?
 - Rebuilding an Index Online
 - How Online Index Operations work
 - Defraging an Index
- Automation!

Why?

- Minimize the Number of Pages Required to Store the Data
- Minimize Cache Requirements
- Improve OLTP Performance
 - Inserts faster – no Splits
 - Updates faster – no relocation/splits
- Improve OLAP Performance – Fewer Pages to Read (Condensed)
- Less Resources Required

When?

- Depends on the Table
 - OLTP – Depends on the level of modifications
 - DSS – Based on build strategy
 - Drop all indexes (NC first, then CL)
 - Load data into HEAP
 - Create all indexes (CL first, then NC, use a Fillfactor of 100)
- Depends on the Usage
 - Large Queries
 - Amount of Cache
- Performance Gains Outweigh the Maintenance Requirements and Potential Inaccessibility (due to Locking) of the Table

Index Maintenance

Table/Index Rebuilds

- SQL Server 2000
 - DBCC DBREINDEX
 - CREATE with DROP_EXISTING
- SQL Server 2005
 - All of the above
 - ALTER INDEX...REBUILD replaces DBCC DBREINDEX
 - DBCC DBREINDEX is OFFLINE only
 - Two modes of rebuilding:
 - ONLINE – allows concurrent user access (queries as well as modifications) to the index during rebuild
 - OFFLINE – works using locks (same as 2000)

Rebuilding an Index

```
ALTER INDEX member_ident  
ON member REBUILD  
WITH (ONLINE = ON, FILLFACTOR = 90)
```

Msg 2725, Level 16, State 2, Line 1

Online index operation cannot be performed for index 'member_ident' because it contains column 'photograph' of data type text, ntext, image, varchar(max), nvarchar(max), varbinary(max) or xml. The operation must be performed offline.

```
ALTER INDEX ChargePK  
ON Charge REBUILD  
WITH (ONLINE = ON, FILLFACTOR = 90)
```

Index Maintenance

ONLINE Rebuild

Indexes can be rebuild ONLINE if:

- The index is NOT disabled
- The index does not include any large data types:

LOB DATA	image	text	ntext
	varbinary(max)	varchar(max)	nvarchar(max)
	xml		

- The index is not a partition
- The index is not on a local temp table
- A disabled nonclustered index that does not have a disabled clustered index CAN be rebuilt online

Index Maintenance

OFFLINE LOB_COMPACTION

- SQL Server 2000
 - Large objects can become fragmented
 - DBCC SHOWCONTIG could show it – with ExtentScanFragmention on IndID 255
 - No commands compact freed space within BLOBs without exporting/importing
 - For more info: Blog entry, Wednesday, August 25, 2004
- SQL Server 2005
 - When a clustered index is rebuilt – all LOB columns within the table are compacted
 - When a non-clustered index that includes LOB data types is rebuild – all LOB columns within the leaf level of the index are compacted

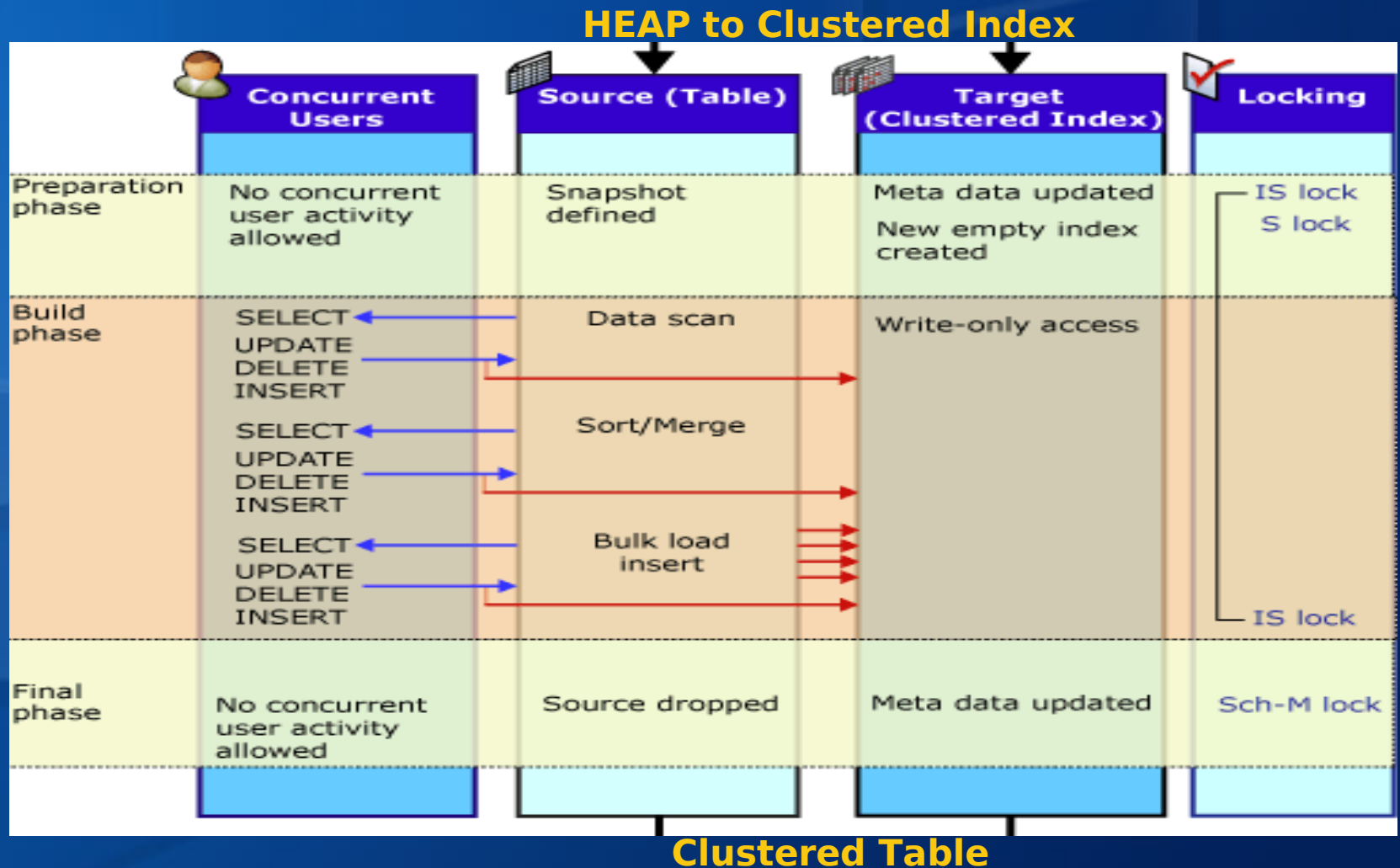
Index Maintenance

How Online Index Rebuilds Work

- Initial index “snapshot” generated
(does not use or require snapshot isolation, however – uses row level versioning in tempdb)
- The “source” index is maintained and queries are directed at “source” for index usage/optimization
- The “target” index is the index being rebuilt and is only used internally for the rebuild process
- Updates are directed at BOTH the source and the target during the index rebuild
- Once rebuilt there's a very slight pause to redirect access ALL requests to target index
- Source index is then dropped in “lazy” fashion

Index Maintenance

How Online Index Rebuilds Work



Online Progress Report

SQL Profiler - [OnlineIndexOps (kltrippw2k3\sql2005idw11)]

File Edit View Replay Tools Window Help

EventClass	TextData	EventSubClass	ObjectID	ObjectName	BigIntData1	BigIntData2	Application
ExistingConnection	-- network protocol: LPC set quoted_id...						SQL Serv
SQL:BatchStarting	alter index test on member rebuild with (...)						SQL Serv
Progress Report: Online Inde...		1 - Start	213575799	test			SQL Serv
Progress Report: Online Inde...		2 - Stage 1 execution begin					SQL Serv
Progress Report: Online Inde...		6 - Inserted row count	213575799		7221	0	SQL Serv
Progress Report: Online Inde...		6 - Inserted row count	213575799		10000	0	SQL Serv
Progress Report: Online Inde...		3 - Stage 1 execution end					SQL Serv
Progress Report: Online Inde...		7 - Done	213575799	test			SQL Serv
SQL:BatchCompleted	alter index test on member rebuild with (...)						SQL Serv
Trace Pause							
Trace Start							

alter index test on member rebuild with (online = on)
go

Trace is running. Ln 11, Col 3 Rows: 29 Connections: 1

Index Maintenance

Special Considerations for ONLINE Rebuild

- Concurrent activity is impacted for very brief period at start and completion of rebuild
- Performance will be impacted as changes are directed to two indexes
- Changing from a HEAP to a Clustered Table or from a Clustered Table to a HEAP causes an internal “mapping index” to be created to allow “conversion” for row lookup between the “source” and “target” indexes
- Only one online index operation per table

Error 1912: Could not proceed with index DDL operation on x because it conflicts with an online operation that is already in progress on the object.

Index Rebuild Tips

- **TIP:** To achieve better base table Availability, you might consider logical/vertical partitioning for some LOB data if base table access (non-LOB columns) must stay ONLINE. This can be especially useful if the LOB data is not always needed.
- **TIP:** To reduce the amount of disk space needed for OFFLINE non-clustered index rebuilds, disable the index before rebuilding.
- **TIP:** To keep your tables ONLINE, do not disable the clustered index.

Disabling an Index

- Prevents access to the index
- All data for non-clustered index is deleted, metadata remains – table remains online when a disabled non-clustered index is rebuilt
- If clustered index disabled, all data is inaccessible and non-clustered indexes are automatically disabled (i.e. dropped)
- A disabled index can be rebuilt or dropped. To enable the index, rebuild it
- DBCC CHECKDB and DBCC INDEXDEFRAG do not act on disabled indexes
- Use ALTER INDEX...REBUILD to rebuild a disabled index

Disabling an Index

Secondary Impact and Rebuilding

- Disabling the index also disables the constraint
 - Foreign Key constraints which reference the disabled constraint will automatically be disabled with warning
 - Foreign Key constraints must be **manually** re-enabled using ALTER TABLE CHECK CONSTRAINT
- Disabling a clustered index also disables the non-clustered indexes on the table
 - Rebuilding the clustered index does NOT rebuild the non-clustered unless ALTER INDEX ALL REBUILD specified
 - Use ALTER INDEX REBUILD to rebuild the non-clustered indexes

Automating Index Rebuilds

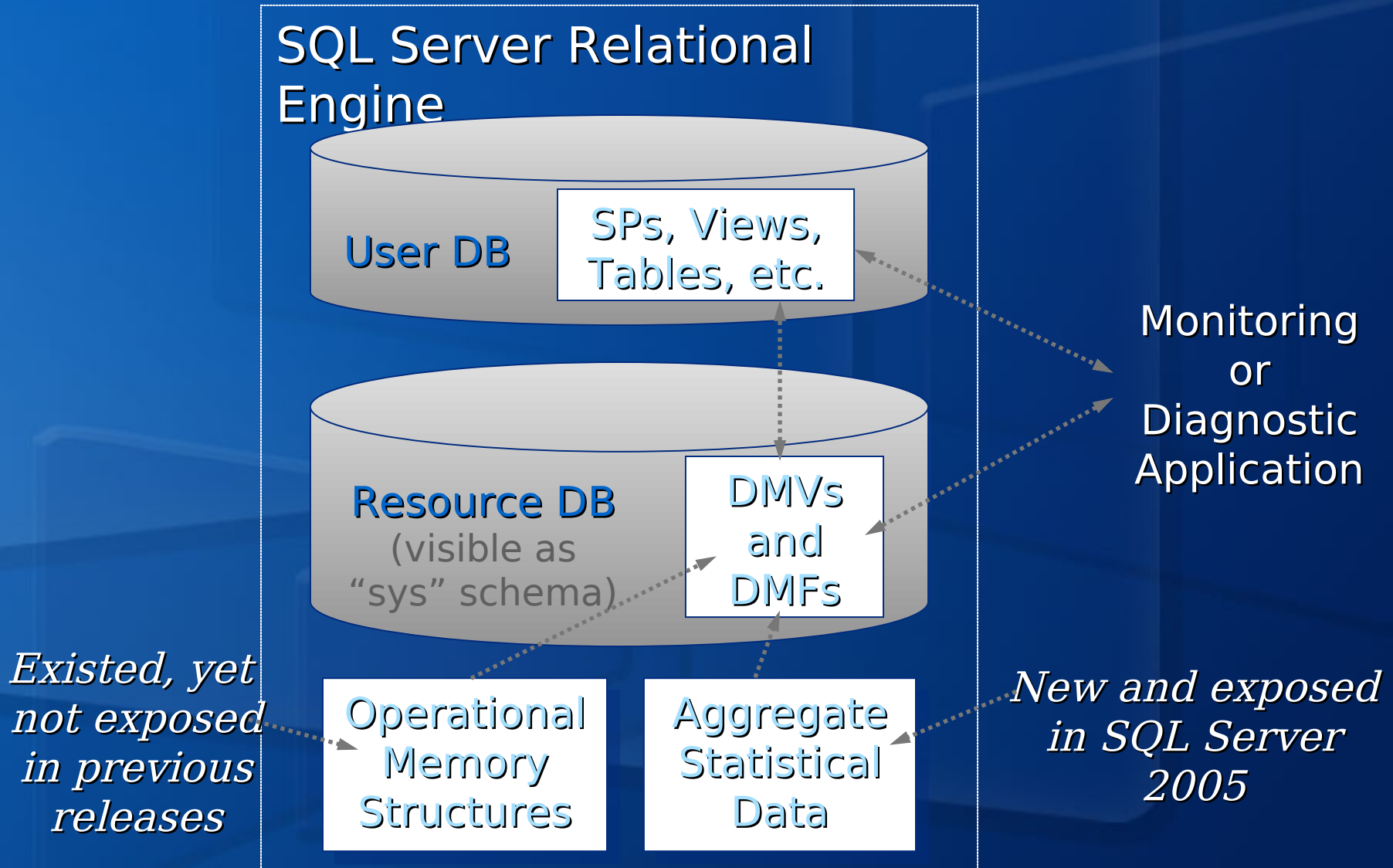
- Use a Loop/Cursor to Walk the Tables
- Programmatically Determine the Level of Fragmentation for Each Index
- Only Rebuild those with more fragmentation than desired
- Create as a Procedure
- Setup a Job!

SQL 2005 Index-related New Features

If time Permits: DMV details

- Dynamic Management Views and the Resource Database
- Object Groupings
- Dynamic Management Objects
- DMObject Scripts to analyze input parameters and output results
- Understanding Query Execution

DM_Object Interface



General Server DM Objects

- **dm_db_***
 - Databases and database objects
- **dm_exec_***
 - Execution of user code and associated connections
- **dm_os_***
 - Memory, locking & scheduling
- **dm_tran_***
 - Transactions & isolation
- **dm_io_***
 - Input/Output on network and disks

Finding DM Objects

Three object types:

- Inline Table-valued Function
- Multi-statement Table-valued Function
- View

```
SELECT so.*  
FROM sys.system_objects AS so  
WHERE so.name LIKE N'dm_%'  
ORDER BY so.type
```

Using DM Objects

- What type of object is it?

```
SELECT dbo.DMObjectType(N'dm_obj_name')
```

- If it's a function – what are the input parameters, their order and type?

```
SELECT *  
FROM dbo.DMFunctionParams(N'dm_obj_name')  
ORDER BY ParameterPosition
```

- What's the result set?

```
SELECT *  
FROM dbo.DMObjectColumns(N'dm_obj_name')  
ORDER BY ColumnPosition
```

Using DM Objects

- Find out all of this with one simple sproc:

EXEC dbo.sp_GetDMObjectInfo N'dm_obj_name'

	DMObjectName	ParameterPosition	ParameterName	DataType	MaxLength
1	dm_db_index_phys...	1	@DatabaseId	smallint	2
2	dm_db_index_phys...	2	@ObjectId	int	4
3	dm_db_index_phys...	3	@IndexId	int	4
4	dm_db_index_phys...	4	@PartitionNumber	int	4
5	dm_db_index_phys...	5	@Mode	nvarchar	40

	DMObjectName	ColumnPosition	ColumnHeader	DataType	MaxLength	Precision
1	dm_db_index_phys...	1	database_id	smallint	2	5
2	dm_db_index_phys...	2	object_id	int	4	10
3	dm_db_index_phys...	3	index_id	int	4	10
4	dm_db_index_phys...	4	partition_number	int	4	10
5	dm_db_index_phys...	5	index_type_desc	nvarchar	120	0
6	dm_db_index_phys...	6	alloc_unit_type_desc	nvarchar	120	0
7	dm_db_index_phys...	7	index_depth	tinyint	1	3
8	dm_db_index_phys...	8	index_level	tinyint	1	3
9	dm_db_index_phys...	9	avg_fragmentation_in_percent	float	8	53
10	dm_db_index_phys...	10	fragment_count	bigint	8	19
11	dm_db_index_phys...	11	avg fragment size in pages	float	8	53

Selected DMVs

- Index-related DMVs
 - `sys.dm_db_index_physical_stats`
 - Size and fragmentation information for tables and indexes
 - `sys.dm_db_index_operational_stats`
 - Internals information for table and index activities
 - `sys.dm_db_index_usage_stats`
 - Index statistics and usage counts information for individual indexes
 - `sys.dm_db_index_partition_stats`
 - Page and row-count information for every partition

Understanding Query Execution

More DMVs to check out!

- **sys.dm_exec_query_stats**

- One row per query plan currently in the cache
- Min, max, avg, last for each: Execution time, Physical reads/writes, Logical reads/writes
- Execution count, first and last execution times
- Number of times query has been recompiled
- Uses a pointer to refer to sql_text (for efficiency)

- **sys.dm_exec_sql_text()**

- In memory SQL text
- Use with sql_text pointer

- **sys.dm_exec_query_plan()**

- In memory Execution and Query Plans
- Use with plan_handle pointer

Chalk Talk (CHT013)
Performance Tuning SQL
Server 2005 using DMVs
Gert Drapers

Thu Jul 7 08:30 - 09:45 Room:
T

Overview

- Everything is about Availability
- How Fragmentation Occurs
- What Fragmentation Means
- How to See Fragmentation
- How to Minimize Fragmentation
- Rebuilding an Index
 - Why
 - When
 - How
- Automating Index Rebuilds
- Understanding Index Usage

Resources

- The SQL Server 2005 Developer Center on msdn
<http://msdn.microsoft.com/SQL/2005/default.aspx>
- “SQL Server 2005 Webcasts” contains links for 20+ webcasts
- “SQL Server Hands-On labs” contains links to online virtual lab environments and lab resources
- “SQL Server 2005 Articles” contains sections for different disciplines and links to 25+ articles/whitepapers
- Keep watching the Developer Center, there are new resources every week!

Resources

- Check out www.SQLskills.com for information about upcoming **SQL Immersion** events, useful links and event scripts.
- Read my blog:
<http://www.SQLskills.com/Blogs/Kimberly/>
- Subscribe to SQLskills:
<http://www.SQLskills.com/login.aspx>
- MPress: *SQL Server 2000 High Availability*
Authors: Allan Hirt with Cathan Cook, Kimberly L. Tripp and Frank McBath
ISBN: 0-7356-1920-4
On the SQLskills.com homepage can download a sample chapter





questions?

Community Resources

- Microsoft Community Resources
<http://www.microsoft.com/communities/default.mspix>
- Non-Microsoft Community Resources
<http://www.microsoft.com/communities/related/default.mspix>
- Newsgroups
Converse online with Microsoft Newsgroups, including Worldwide
<http://www.microsoft.com/communities/newsgroups/default.mspix>
- User Groups
Meet and learn with your peers
<http://www.microsoft.com/communities/usergroups/default.mspix>
- Attend a free chat
<http://www.microsoft.com/communities/chats/default.mspix>
- Attend a free web cast
<http://www.microsoft.com/usa/webcasts/default.asp>
- Most Valuable Professional (MVP)
<http://mvp.support.microsoft.com/>

Microsoft Learning Resources:

Get ready for Microsoft Visual Studio 2005 and Microsoft SQL Server 2005 with free Assessments and E-Learning, and a chance to win a laptop, GPS, and more

- Click here to access free Microsoft Learning Assessments
<http://www.microsoft.com/learning/assessment/ind/default.asp>
- To access free Microsoft Learning E-Learning visit
<http://www.microsoft.com/learning/access> and reference the promotional code **9185-TECHED-6650**.

Special offers on Microsoft Certification from Microsoft Learning

- Click here to take advantage of these special offers:
<http://www.microsoft.com/learning/mcp/>

The Microsoft Guide to Support Options

Free of charge support

1

Self Support

- Knowledge Base
- Microsoft Newsgroups
- Support Webcasts

Fee based support

2

Assisted Support

- Email Support
- Phone Support
- Advisory Services
 - Remotely delivered, hourly fee-based, consultative support option

Managed support

3

Contract-based Support

- Microsoft Premier Support
 - Provides premium level support and a designated Account Manager
- Microsoft Essential Support
 - Prepackaged support options including phone and online access to a pool of skilled Technical Account Specialists
- Microsoft Partner Advantage
 - Partner Advantage provides technology partners the best response times and highest level of problem resolution support

Resources:

Microsoft Support Website: <http://support.microsoft.com>

Microsoft TechNet program: <http://technet.microsoft.com>

Microsoft MSDN program: <http://msdn.microsoft.com>



All attendees who submit a session feedback form within 120 minutes after the session ends will have the chance to win an **MPx220** With **Windows Mobile™ Software.**

Added Bonus!! This year you will also have the chance to win one of eight Xbox 360s! We will be giving away two a day – one for the morning sessions, and one for the afternoon sessions.

Good luck!



The Microsoft logo is displayed in a bold, italicized, white sans-serif font. It features a registered trademark symbol (®) at the end. The logo is centered horizontally and has a subtle drop shadow, giving it a three-dimensional appearance against the blue background.

Microsoft®

© 2005 Microsoft Corporation. All rights reserved. This presentation is for
informational purposes only.
MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS SUMMARY.